Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Learning Markov Jump Affine Systems via Regression Trees for MPC

Francesco Smarra and Alessandro D'Innocenzo

Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, L'Aquila, Italy e-mail: {francesco.smarra,alessandro.dinnocenzo}@univaq.it.

Abstract: Model Predictive Control is a well consolidated technique to design optimal control strategies, leveraging the capability of a mathematical model to predict a system's behavior over a time horizon. However, building physics-based models for complex large-scale systems can be cost and time prohibitive. To overcome this problem we propose a methodology to exploit Regression Trees technique in order to build a Markov Jump System model of a large-scale system using historical data, and apply Model Predictive Control. A comparison with an optimal benchmark and related techniques is provided on an energy management system to validate the performance of the proposed methodology.

Keywords: Regression Trees, Model Predictive Control, Switching Systems, Markov Jump Systems, Data-driven.

1. INTRODUCTION

Control of complex cyber-physical systems received an increasing attention in the last years (Lun et al., 2019). Model Predictive Control (MPC) is a well known control strategy used to design optimal control actions to optimize a performance metric while guaranteeing a desired system behavior. To provide such an optimal control strategy, MPC leverages a mathematical model to predict system's behavior over a finite time horizon. However, creating a physics-based model for large-scale systems is often cost and time prohibitive (Smarra et al., 2018a). To overcome this issue a possibility is to use identification techniques to create models using historical data available from the system. Several works deal with this problem, and use both system identification from control theory and machine learning algorithms from computer science to construct models to be used for control applications. To the best of the authors' knowledge, the use of Regression Trees (RT) with predictive control purposes has been addressed for the first time in (Behl et al., 2016), and then extended in (Jain et al., 2017, 2018), where the authors proposed a RT-based strategy that implements Model Predictive Control (MPC) over a horizon of arbitrary length. The aforementioned approaches make use of data-driven *static* models, where the input-output relation is represented by static affine functions instead of dynamical models: such modeling framework neglects the presence of the internal state evolution and loses the information of the past inputs applied to the system over the predictive horizon. In (Smarra et al., 2018b) the authors propose a method to identify a (deterministic) Switching Affine dynamical modeling of a system using historical data, by appropriately adapting the RT algorithm. In (Smarra et al., 2018a) the authors also show on a building automation experimental setup that, in such modeling framework, the knowledge of the forecast of the future disturbance signal can greatly improve the MPC performance. In that case the disturbance consists of weather conditions, which clearly affect the thermal dynamics of a building. However, in many applications the disturbance forecast is not available. Thus, as the main contribution of this paper, we provide a novel methodology to build a Markov Jump System (MJS) (Costa et al., 2006) that identifies the dynamics of the disturbance as a Markov Chain model exploiting historical data. The resulting model can be used to implement Stochastic MPC via standard techniques (Bernardini and Bemporad, 2012). We validate our approach on a benchmark consisting of a bilinear model of a building with 12 states, 4 inputs and 8 disturbances, whose parameters were identified using experiments on a building in Switzerland (Oldewurtel, 2011). We compare the performance of our technique with a baseline method for switching ARX identification (k-LinReg) (Lauer, 2013), with the deterministic approach in (Smarra et al., 2018b) (which considers both the cases of full and no knowledge of the disturbance), and with an *oracle* with perfect knowledge of both the building bilinear dynamical model in (Oldewurtel, 2011) and the future disturbance variables (i.e. perfect weather forecast). Simulations show that our approach outperforms, in terms of model accuracy and control performance the k-LinReg and the deterministic methodologies with no knowledge of disturbance. ***Paper organization.*** Section 2 defines the problem formulation. Section 3 provides a background on the RT-based deterministic modeling framework developed in (Smarra et al., 2020). Section 4 presents our novel methodology to derive a Markov Jump System model using RT, and setup a Model Predictive Control problem. Section 5 provides simulation results.

2. PROBLEM FORMULATION

Let a dataset $\mathcal{D}$ be collected from a physical system, where $\mathcal{D} = \{(y(k), u(k), d(k))\}_{k=1}^{\ell}$ is a finite set of cardinality

$|\mathcal{D}| = \ell$ samples obtained from measurements of, respectively, output signals $y(k) \in \mathbb{R}^n$, control signals $u(k) \in \mathbb{R}^m$ and disturbance signals $d(k) \in \mathbb{R}^p$. The contribution of this paper is to identify a black-box switching model driven by a Markov chain, i.e. a Markov Jump System, with the aim of applying stochastic MPC. In (Smarra et al., 2018b, 2020) we proposed a procedure to derive a predictive model for the future time horizon $j = 0, \ldots, N-1$ as follows:

$$x(k+j+1) = A_{\sigma_j(x(k),d(k))}x(k+j) + \\ + B_{\sigma_j(x(k),d(k))}u(k+j) + f_{\sigma_j(x(k),d(k))} \quad (1)$$

where $x(k) \doteq [y^\top(k) \ \cdots \ y^\top(k-\delta_y) \ u^\top(k-1) \ \cdots \ u^\top(k-\delta_u)]^\top$ is an extended state to characterize an ARX model, $\sigma_j : \mathbb{R}^{n(\delta_y+1)+m\delta_u+p} \to \mathcal{M}_j \subset \mathbb{N}$ associates (via a rectangular partition, as will be illustrated in the following sections) to each pair $(x(k), d(k))$, and each future time step j, an operating mode in a finite set $\mathcal{M}_j$, and δ_y, δ_u are nonnegative scalars denoting the number of autoregressive terms used for the corresponding variables. Since the switching sequence of $\sigma_j(x(k), d(k))$, being a function of $(x(k), d(k))$, is available at step k for any $j = 0, \ldots, N-1$, Equation (1) can be used in the standard formulation of N-steps linear MPC. The optimal solution at each step k can be computed via Quadratic Programming (QP). This approach has also been validated on a Structural Health Monitoring case study Di Girolamo et al. (2020). In many practical cases, the knowledge/forecast at each time k of the future disturbance signal $(d(k+1), \ldots, d(k+N-1))$ can greatly improve the MPC performance, as in the building automation setup we addressed in (Smarra et al., 2018a) where the disturbance consisted of weather conditions. As discussed in (Smarra et al., 2020), it is straightforward to derive the predictive model (1) with $\sigma_j(x(k), d(k), \ldots, d(k+j))$, $j = 0, \ldots, N-1$: using the forecast $(d(k+1), \ldots, d(k+N-1))$ it is possible to predict the future switching sequence and solve MPC via QP. However, if the forecast of the disturbance signal is not available, the sequence $\sigma_j(x(k), d(k), \ldots, d(k+j))$, $j = 0, \ldots, N-1$, can arbitrarily assume $\prod_{j=0}^{N-1} |\mathcal{M}_j|$ values, and the MPC problem turns into a MILP. As the main contribution of this paper we address such problem in Section 4, where we propose an optimization algorithm to derive transition probabilities that characterize the switching sequence as a Markov Chain: this makes the solution of the MPC problem computationally feasible leveraging the theory of Markov Jump Systems (Bernardini and Bemporad, 2012).

Due to space limitation, we refer the reader for more details on the CART algorithm of RT to the Appendix of Smarra et al. (2020), and for more details to the original book Breiman (2017).

3. SWITCHING ARX IDENTIFICATION USING RT

Let a dataset $\mathcal{D} = \{(y(k), u(k), d(k))\}_{k=1}^{\ell}$ be given as defined in Section 2. Let us assume, without loss of generality and for simplicity of presentation, that we wish to predict the value of the scalar variable $y_1(k+1)$ given measurements at time k of the vector $[y^\top(k) \ u^\top(k) \ d^\top(k)]^\top \in \mathbb{R}^{n+m+p}$. The RT algorithm creates a tree structure $\mathcal{T}$ by partitioning the set $\mathcal{D}$ into subsets $\mathcal{D}_i$. Let $|\mathcal{T}|$ denote the number of leaves obtained from the partitioning, then each leaf $i = 1, \ldots, |\mathcal{T}|$ contains a certain number of samples from $\mathcal{D}$ belonging to the hyper-rectangular region R_i, i.e. $\mathcal{D}_i = \{(y(k_1), u(k_1), d(k_1)), \ldots, (y(k_\epsilon), u(k_\epsilon), d(k_\epsilon))\}$, at time instants $k_1, \ldots, k_\epsilon$ that are not necessarily adjacent. The algorithm associates to each leaf $\mathcal{D}_i$ a prediction

$$\hat{y}_{1,i}(k+1) = \frac{\sum_{(y(k),u(k),d(k))\in\mathcal{D}_i} y_1(k+1)}{|\mathcal{D}_i|} \quad (2)$$

as the average of the response values associated to each sample in $\mathcal{D}_i$.

From now on, for ease of reading, we remove the "hat" from the estimated model variables such as $\hat{y}$ in (2). The difference with the measured variables y in the dataset will be clear from the context. The idea is to create nN predictive trees $\mathcal{T}_{\iota,j}$, $\iota = 0, \ldots, n$, $j = 0, \ldots, N-1$, each one to predict $y_\iota(k+j+1)$ over the N steps of the horizon, and replace the average response given by (2) associated to each leaf of each tree with an LTI model in order to obtain the following predictive model $\forall\, j = 0, \ldots, N-1$ to describe the dynamics' evolution over the horizon

$$x(k+j+1) = A_{\sigma_j(x(k),d(k))}x(k+j) + \\ + B_{\sigma_j(x(k),d(k))}u(k+j) + f_{\sigma_j(x(k),d(k))}, \quad (3)$$

where $x(k) \doteq [y^\top(k) \ \cdots \ y^\top(k-\delta_y) \ u^\top(k-1) \ \cdots \ u^\top(k-\delta_u)]^\top$ is an extended state to characterize a switching ARX model, and $\sigma_j : \mathbb{R}^{n(\delta_y+1)+m\delta_u+p} \to \{1, \ldots, |\mathcal{T}_{\iota,j}|\}$ is a switching signal that associates to each pair $(x(k), d(k))$ n leaves of $\mathcal{T}_{\iota,j}$, $\iota = 1, \ldots, n$.

To this aim, we first construct an extended dataset $\mathcal{X} \doteq \{(x(k), u(k), d(k))\}_{k=1}^{\ell}$. We partition such dataset in two disjoint sets: $\mathcal{X}_c = \{u(k)\}_{k=1}^{\ell}$ of data associated to the control variables, and $\mathcal{X}_{nc} = \{(x(k), d(k))\}_{k=1}^{\ell}$ of data associated to non-control variables. Then, we apply the CART algorithm only on $\mathcal{X}_{nc}$; thus, we create nN trees $\{\mathcal{T}_{\iota,j}\}$, each constructed to predict the variable $y_\iota(k+j+1)$. In particular, we associate to each leaf ι, i_j, corresponding to the partition $\mathcal{X}_{nc,\iota,i_j}$, of each tree $\mathcal{T}_{\iota,j}$ the following affine model

$$x_\iota(k+j+1) = A'_{\iota,i_j}x(k) + \sum_{\alpha=0}^{j} B'_{\iota,i_j,\alpha}u(k+\alpha) + f'_{\iota,i_j}, \quad (4)$$

where the coefficients of matrices A'_{ι,i_j}, $B'_{\iota,i_j,\alpha}$ and f'_{ι,i_j} are obtained in each leaf ι, i_j by fitting the corresponding set of samples via the classical Least Squares method, as illustrated in Problem 2 of (Smarra et al., 2020). From (4) we can easily construct the following affine model to be used in the MPC formulation by combining for $\iota = 1, \ldots, n$ the matrices of each leaf ι, i_j, i.e. $\forall i_j$, $\forall j$

$$x(k+j+1) = A'_{i_j}x(k) + \sum_{\alpha=0}^{j} B'_{i_j,\alpha}u(k+\alpha) + f'_{i_j}. \quad (5)$$

The following proposition shows how, given a model as in (5), it is possible to construct a model as in (3) that is equivalent to it for any initial condition, any switching sequence, and any control sequence.

Proposition 1. (Smarra et al., 2020) Let A'_{i_j}, $B'_{i_j,\alpha}$ and f'_{i_j}, $\alpha = 0, \ldots, j$, $j = 0, \ldots, N-1$, be given. If A'_{i_j} is invertible for $j = 0, \ldots, N-1$, then there exists a model in the form $\bar{x}(k+j+1) = A_{i_{j-1},i_j}\bar{x}(k+j) + B_{i_{j-1},i_j}u(k+j) + f_{i_{j-1},i_j}$

such that for any initial condition $\bar{x}(k) = x(k) = x_k$, any switching sequence $i_0, \ldots, i_{N-1}$ and any control sequence $u(k), \ldots, u(k+N-1)$, then $\bar{x}(k+j+1) = x(k+j+1)$, $\forall j = 0, \ldots, N-1$.

Remark 2. To be precise, in Smarra et al. (2020), we have shown that Proposition 1 holds in the particular case of $\delta_u = 0$. However, this is without any loss of generality, since in the case of $\delta_u \neq 0$ we can still make use of the switching ARX model (5) in the MPC problem formulation.

The obtained model can be used to formalize the following:

Problem 3. (Model Predictive Control)

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & x_{k+N}^\top Q_N x_{k+N} + \sum_{j=0}^{N-1} \left(x_{k+j}^\top Q x_{k+j} + u_{k+j}^\top R u_{k+j} \right) \\
\text{subject to} \quad & x_{k+j+1} = A_{i_{j-1},i_j} x_{k+j} + B_{i_{j-1},i_j} u_{k+j} + f_{i_{j-1},i_j} \\
& x_{k+j} \in \mathcal{O},\ u_{k+j} \in \mathcal{U},\ x_{k+N} \in \mathcal{O}_N \\
& x_k = x(k),\ j = 0, \ldots, N-1,
\end{aligned}$$

where $\mathcal{O}, \mathcal{U}, \mathcal{O}_N$ are polyhedra that specify the variables constraints. At any time k we can use the measurements $(x(k), d(k))$ to determine the switching sequence $i_0, \ldots, i_{N-1}$, hence characterizing $A_{i_{j-1},i_j}, B_{i_{j-1},i_j}, f_{i_{j-1},i_j}$ in Problem 3, which can be solved as in classical MPC.

4. MARKOV SARX IDENTIFICATION USING RT

In many practical cases, the knowledge at each time k of the future disturbance signal $(d(k+1), \ldots, d(k+N-1))$ can greatly improve the MPC performance, as in the building automation setup we addressed in (Smarra et al., 2018a), where the disturbance consisted of weather conditions. In that case, we assumed to have knowledge of weather forecast, and derived a dynamical model where the switching signal also depended on the future disturbance signal, i.e. $\sigma_j(x(k), d(k), \ldots, d(k+j)), \forall j = 0, \ldots, N-1$. In the technique described in Section 3, this can be easily done be appropriately redefining the dataset as $\mathcal{X} = \{(x(k), u(k), d(k), \ldots, d(k+N-1))\}_{k=1}^{\ell}$. At each time k of the run-time solution of Problem 3, the switching sequence $i_0, \ldots, i_{N-1}$ depends on the future disturbances: if these are known, Problem 3 can still be solved using QP; if the future disturbances are unknown, the sequence $i_0, \ldots, i_{N-1}$ can (non-deterministically) assume any value within a finite set, and Problem 3 becomes a MILP. To tackle this problem, one can extract a predictive model of the switching signal $i_0, \ldots, i_{N-1}$ by means of a Markov Chain, exploiting the historical data. In the following, we illustrate how to modify the algorithm proposed in Section 3 to derive a Markov Jump System (Costa et al., 2006) that takes into account the probabilistic jumps between leaves. The resulting model can be used to implement Stochastic MPC via standard algorithms (Bernardini and Bemporad, 2012).

Let us consider a predictive horizon equal to N, and our dataset $\mathcal{X} \doteq \{(x(k), u(k), d(k), \ldots, d(k+N-1))\}_{k=1}^{\ell}$. Using the technique illustrated in the previous section, we create a model as in (3) using, for each predictive step j, the dataset $\mathcal{X}_{nc,j} \doteq \{s_j(k)\}_{k=1}^{\ell}$, with $s_j(k) \doteq (x(k), d(k), \ldots, d(k+j))$: in fact, by causality, the switching signal at time $k+j$ can only depend on the disturbance up to time $k+j$, i.e. $\sigma_j(x(k), d(k), \ldots, d(k+j))$.

As illustrated above, when using such model to solve the MPC problem at time k, the switching sequence depends on future unknown disturbances: to overcome this problem we derive predictive models for $j = 0, \ldots, N-1$ by

$$x(k+j+1) = A'_{\theta(k+j)} x(k) + \sum_{\alpha=0}^{j} B'_{\theta(k+j),\alpha} u(k+\alpha) + f'_{\theta(k+j)}, \quad (6)$$

where $\theta(k)$ is a Markov Chain that drives the switching rule among the leaves of all trees. Note that, given model (6) and using the same approach as in Proposition 1, it is straightforward to derive a Markov Jump System model for $j = 0, \ldots, N-1$ by

$$x(k+j+1) = A_{\theta(k+j)} x(k+j) + B_{\theta(k+j)} u(k+j) + f_{\theta(k+j)}. \quad (7)$$

To fully characterize $\theta(k)$ we need to define the initial state θ_0 and a transition probability matrix (TPM) P. The initial state is given by $\theta_0 = i_0$, i.e. by the leaf i_0 such that $s_0(k) \in R_{i_0}$: this assignment can only be done in run-time at any time step k using the measurement of $s_0(k) = (x(k), d(k))$.

To define the TPM P we need to compute, for any pair of trees $\mathcal{T}_j$ and $\mathcal{T}_{j+1}$, with $j = 0, \ldots, N-2$, the transition probability $p(i_j, i_{j+1})$ from each leaf i_j of $\mathcal{T}_j$ to each leaf i_{j+1} of $\mathcal{T}_{j+1}$. We propose two methods to derive P: the first one is a *naive* method which leverages the trees used to construct the predictive models. Let us denote, with a slight abuse of notation, by $\mathcal{T}_j(s_j(k))$ the leaf of $\mathcal{T}_j$ that contains the sample $s_j(k)$. Let $|i_j|$ be the number of samples in the leaf i_j of $\mathcal{T}_j$ and $n(i_j, i_{j+1})$ be the number of samples that jump from leaf i_j to leaf i_{j+1}, i.e. the number of samples $s_j(k) \in i_j$ such that $\mathcal{T}_{j+1}(s_{j+1}(k)) = i_{j+1}$ (we recall that $s_{j+1}(k) = (x(k), d(k), \ldots, d(k+j), d(k+j+1)) = (s_j(k), d(k+j+1)))$. Then we define

$$p(i_j, i_{j+1}) \doteq n(i_j, i_{j+1}) \cdot |i_j|^{-1}. \quad (8)$$

The problem of this approach is that, even thought the jump probabilities are necessarily given by (8), the trees $\mathcal{T}_j$ partition the dataset to minimise the estimation error of a deterministic prediction of $x(k+j+1)$, $j = 0, \ldots, N-1$ given by (5), and not to minimise the expected value of (6), which involves the jump probabilities:

$$\begin{aligned}
&\mathbb{E}[x(k+j+1) \mid x(k), d(k), u(k), \ldots, u(k+j)] \quad (9)\\
&= \sum_{\theta} \mathbb{P}[\theta(k+j) = \theta \mid \theta(k) = \mathcal{T}_0(s_0(k))] \cdot \\
&\cdot (A'_\theta x(k) + \sum_{\alpha=0}^{j} B'_{\theta,\alpha} u(k+\alpha) + f'_\theta)
\end{aligned}$$

To address this problem we propose a different approach where, on the basis of the trees $\mathcal{T}_{j+1}$, $j = 0, \ldots, N-2$ derived as described in the previous section, we construct $N-1$ new trees Π_j, $j = 0, \ldots, N-2$. We will provide a construction technique of the trees Π_j by appropriately extending the dataset and choosing the variable to be predicted: we will show that, if the computation of the transition probabilities $p(\pi_j, i_{j+1})$ from each leaf π_j of Π_j to each leaf i_{j+1} of $\mathcal{T}_{j+1}$ is done as in Equation (8), we minimize the mean square estimation error w.r.t. the conditional expectation (9).

Let us first define a new data set $\mathcal{Z}_j = \{z_j(k)\}_{k=1}^{\ell}$, extracted from $\mathcal{X}$, for $j = 0, \ldots, N-2$, as follows:

$$z_j(k) \doteq A'_{\mathcal{T}_j(s_j(k))} x(k) + \sum_{\alpha=0}^{j} B'_{\mathcal{T}_j(s_j(k)),\alpha} u(k) + f'_{\mathcal{T}_j(s_j(k))}.$$

Basically, $z_j(k)$ is the prediction of $x(k+j+1)$ given $s_j(k), u(k), \ldots, u(k+j)$ given by (5), which coincides with the conditional expectation (9) where we also assume knowledge of future disturbances, i.e. $z_j(k) = \mathbb{E}[x(k+j+1) \mid s_j(k), u(k), \ldots, u(k+j)]$: indeed the knowledge of $s_j(k)$ allows to deterministically determine $\mathcal{T}_j(s_j(k))$.

Let us now consider the dataset $\mathcal{Z}_j \cup \mathcal{X}$ and assume to grow $N-1$ trees Π_j, each applying the CART algorithm restricted to $\mathcal{X}_{nc,0}$, and choosing $z_j(k)$ as the variable to predict. In each leaf π_j of Π_j, the CART solves the following optimization problem (see CART algorithm description in the Appendix of Smarra et al. (2020)):

$$\min_{\jmath,\xi} \left[\min_{c_L} \sum_{s_0(k) \in R_L(\jmath,\xi)} (z_j(k) - c_L)^2 + \min_{c_R} \sum_{s_0(k) \in R_R(\jmath,\xi)} (z_j(k) - c_R)^2 \right] \tag{10}$$

Given the optimal choice of $\jmath^*$ and ξ^*, the inner minimization in Equation (10) splits the original leaf into two leaves R_L and R_R up to the end of the CART algorithm. Let c_{π_j} be the optimal estimation obtained via (10) in any leaf π_j of the tree Π_j obtained from the CART algorithm. We show that, if the transition probabilities $p(\pi_j, i_{j+1})$ are computed as in Equation (8), then c_{π_j} is equal to the conditional expectation of the prediction given by (9), i.e. $\mathbb{E}[x(k+j+1) \mid s_0(k), u(k), \ldots, u(k+j)]$:

$$\begin{aligned}
c_{\pi_j} &= \sum_{s_0(k) \in \pi_j} \frac{z_j(k)}{|\pi_j|} \\
&= \sum_{s_0(k) \in \pi_j} \frac{A'_{\mathcal{T}_{j+1}(s_j(k))} x(k) + \sum_{\alpha=0}^{j} B'_{\mathcal{T}_{j+1}(s_j(k)),\alpha} u(k+j) + f'_{\mathcal{T}_{j+1}(s_j(k))}}{|\pi_j|} \\
&= \sum_{i_{j+1}} \sum_{\substack{s_0(k) \in \pi_j, \\ s_j(k) \in i_{j+1}}} \frac{A'_{i_{j+1}} x(k) + \sum_{\alpha=0}^{j} B'_{i_{j+1},\alpha} u(k+j) + f'_{i_{j+1}}}{|\pi_j|} \\
&= \sum_{i_{j+1}} \sum_{\substack{s_0(k) \in \pi_j, \\ s_{j+1}(k) \in i_{j+1}}} \frac{\mathbb{E}[x(k+j+1) \mid s_{j+1}(k), u(k), \ldots, u(k+j)]}{|\pi_j|} && (11) \\
&= \sum_{i_{j+1}} \frac{n(\pi_j, i_{j+1})}{|\pi_j|} \cdot \mathbb{E}[x(k+j+1) \mid s_{j+1}(k), u(k), \ldots, u(k+j)] && (12) \\
&= \sum_{i_{j+1}} p(\pi_j, i_{j+1}) \cdot \mathbb{E}[x(k+j+1) \mid s_{j+1}(k), u(k), \ldots, u(k+j)] && (13) \\
&\cong \mathbb{E}[x(k+j+1) \mid s_0(k), u(k), \ldots, u(k+j)]. && (14)
\end{aligned}$$

Note that in Equations (11), (12) and (13) the terms inside the expectation are deterministic: we use the conditional expectation formulation just to emphasise the dependence on the variables assumed to be known. In (14) we assume that the dataset $\mathcal{X}$ consists of independently drawn observations, and that the number of samples in each region of the tree is large enough to neglect the Standard Error of the sample Mean (SEM): as a consequence, the expectation can be assumed approximately equal to the sample mean. In conclusion, running the CART algorithm on our extended dataset minimizes the square of the error between the samples of the dataset $\mathcal{X}$ and the corresponding conditional expectation of the predictive model (6).

Using the previous two approaches, we can construct a transition probability matrix P as in (15)

$$P = \begin{bmatrix} \mathbf{0} & P_{0,1} & 0 & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & P_{1,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & P_{N-2,N-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I \end{bmatrix}, \tag{15}$$

where $P_{j,j+1} = [p(i_j, i_{j+1})]$ or $P_{j,j+1} = [p(\pi_j, i_{j+1})]$, according to the used methods as discussed above. Note that such model only provides the system's dynamics for a future horizon of N time steps, which is enough to implement an MPC with predictive horizon of N steps. For this reason the transition probabilities starting from the leaves of $\mathcal{T}_N$ are irrelevant for the solution of the MPC optimization problem, and can thus be chosen arbitrarily, as for example the identity matrix in the last row of (15).

4.1 Stochastic MPC setup via RT

The Markov Jump System model (7) can be thus used to formalize the following MPC problem.

Problem 4.

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & \mathbb{E}\left[x_{k+N}^\top Q_N x_{k+N} + \sum_{j=0}^{N-1} \left(x_{k+j}^\top Q x_{k+j} + u_{k+j}^\top R u_{k+j} \right) \right] \\
\text{subject to} \quad & x_{k+j+1} = A_{\theta(k+j)} x_{k+j} + B_{\theta(k+j)} u_{k+j} + f_{\theta(k+j)} \\
& \mathbb{E}[x_{k+j}] \in \mathcal{O}, u_{k+j} \in \mathcal{U}, \mathbb{E}[x_{k+N}] \in \mathcal{O}_N \\
& x_k = x(k),\ j = 0, \ldots, N-1,
\end{aligned}$$

where $\mathcal{O}, \mathcal{U}, \mathcal{O}_N$ are polyhedra that specify the variables constraints. Note that, differently from the previous sections, in this case we do not need to determine the future switching sequence at step k, but only the initial state θ_0 of $\theta(k)$ and its transition probability matrix P. The solution of Problem 4 can be computed via standard algorithms (Bernardini and Bemporad, 2012). Algorithm 1 summarizes the whole procedure illustrated in this section.

Remark 5. As discussed in the previous section, an additional outcome of our method is that important properties of the dynamical systems (7) can be characterized and verified using several classical techniques available in the literature, e.g. for stability, stabilizability, controllability and observability (Costa et al., 2006), as well as for stability and recursive feasibility related to Problem 4 (Bernardini and Bemporad, 2012).

5. CASE STUDY

In this section we compare the proposed stochastic approaches with their deterministic counterparts both in terms of model accuracy and control performance. As a benchmark we consider a bilinear building model developed at the Automatic Control Laboratory at ETH Zurich. It captures the essential dynamics governing the zone-level operation while considering the external and the internal thermal disturbances. By Swiss standards, the model used for this study is of a heavyweight construction with a high window area fraction on one facade and high internal gains due to occupancy and equipments (Gyalistras and Gwerder, 2010). As mentioned above, our methodology fits well for large-scale systems where identifying a physics-based mathematical model can be prohibitive. However, in order to validate our methodology, we provide a comparison with an optimal MPC benchmark considering the

Algorithm 1 Data-driven Stochastic MPC with RT

<u>DESIGN TIME: OFFLINE</u>
INPUT: DATASET $\mathcal{X} = \{(x(k), u(k), d(k)), \ldots, d(k+j)\}_{k=1}^{\ell}$
procedure TRAINING LTI MODELS IN LEAVES
  Compute matrices A_{i_{j-1},i_j}, B_{i_{j-1},i_j}, f_{i_{j-1},i_j}, $\forall (i_{j-1}, i_j)$ of Proposition 1;
  Generate dataset $\mathcal{Z}_j = \{z_j(k)\}_{k=1}^{\ell}$, $j = 0, \ldots, N-2$;
  Build $N-1$ trees Π_j using $\mathcal{X}_{nc,0}$;
  for all $j = 0, \ldots, N-2$ **do**
    Compute $p(i_j, i_{j+1})$ (resp. $p(\pi_j, i_{j+1})$) using the trees $\mathcal{T}_j$ (resp. Π_j) and $\mathcal{T}_{j+1}$ using the naive (resp. optimal) method;
  end for
  Compute transition probability matrix P using (15);
end procedure

<u>RUN TIME: ONLINE</u>
INPUT: MATRICES $A_{i_{j-1},i_j}, B_{i_{j-1},i_j}, f_{i_{j-1},i_j}$, $\forall (i_{j-1}, i_j)$, MATRIX P, CONSTRAINT SETS $\mathcal{O}$, $\mathcal{U}$, $\mathcal{O}_N$, WEIGHT MATRICES Q_N, Q, R
procedure STOCHASTIC MPC VIA SA
  while $k \geq 0$ **do**
    Using $(x(k), d(k))$ determine initial state $\theta_0 = i_0$ of $\theta(k)$;
    Using θ_0, P, and A_{i_{j-1},i_j},B_{i_{j-1},i_j},f_{i_{j-1},i_j}, $\forall (i_{j-1}, i_j)$ solve Problem 4 to determine optimal inputs $u_k^*, \ldots, u_{k+j}^*$;
    Apply the first input $u(k) = u_k^*$;
  end while
end procedure

bilinear model for which the physics-based dynamics are known a priori. To this end, we build 4 types of dynamical models – deterministic Switching Affine with knowledge of disturbance forecast (SA w/ forecast), deterministic Switching Affine without any knowledge of disturbance forecast (SA w/o forecast), naive Markov Jump System (nMJS), and optimal Markov Jump System (oMJS) – by generating data using the bilinear model. In the model validation procedure, we also compare the prediction accuracy of such models against a baseline method for Switched ARX identification (Lauer, 2013), showing better performance of our methods in the considered case study.

Model description. The bilinear model has 12 internal states including the inside zone temperature $\mathsf{T}_{\text{in}} \in \mathbb{R}$, the slab temperatures $\mathsf{T}_{\text{sb}} \in \mathbb{R}^5$, the inner wall $\mathsf{T}_{\text{iw}} \in \mathbb{R}^3$ and the outside wall temperature $\mathsf{T}_{\text{ow}} \in \mathbb{R}^3$. The state vector is defined as $x := [\mathsf{T}_{\text{in}}\ \mathsf{T}_{\text{sb}}^\top\ \mathsf{T}_{\text{iw}}^\top\ \mathsf{T}_{\text{ow}}^\top]^\top$. There are 4 control inputs including the blind position B, the gains due to electric lighting L, the evaporative cooling usage factor C, and the heat from the radiator H such that $u := [\mathsf{B}\ \mathsf{L}\ \mathsf{C}\ \mathsf{H}]^\top$. B and L affect both room illuminance and temperature due to heat transfer, whereas C and H affect only the temperature. The model is subject to 5 weather disturbances: solar gains with fully closed blinds Q_{sc} and with open blinds Q_{so}, daylight illuminance with open blinds I_{o}, external dry-bulb temperature T_{db} and external wet-bulb temperature T_{wb}. The hourly weather forecast, provided by MeteoSwiss, was updated every 12 hrs. Therefore, to improve the forecast, an autoregressive model of the uncertainty was considered. Other disturbances come from the internal gains due to occupancy Q_{io} and due to equipments Q_{ie} which were assumed as per the Swiss standards (Merkblatt, 2006). We define $d := [\mathsf{Q}_{\text{sc}}\ \mathsf{Q}_{\text{so}}\ \mathsf{I}_{\text{o}}\ \mathsf{Q}_{\text{io}}\ \mathsf{Q}_{\text{ie}}\ \mathsf{T}_{\text{db}}\ \mathsf{T}_{\text{wb}}]^\top$. For further details, we refer the reader to (Oldewurtel, 2011). The model dynamics are given below, where the bilinearity is present in both input-state and input-disturbance, and $A \in \mathbb{R}^{12\times 12}$, $B_{xu,i} \in \mathbb{R}^{12\times 12}$, $B_{du,i} \in \mathbb{R}^{12\times 7}$, $\forall i = 1, 2, 3, 4$:

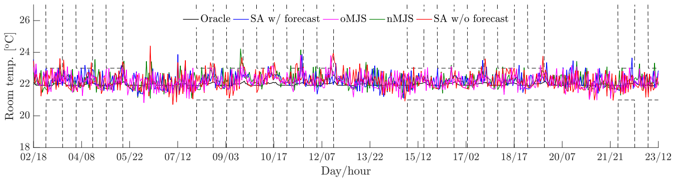

Fig. 1. Controlled temperature from January 2nd to January 23rd.

$$x(k+1) = Ax(k) + (B_u + B_{xu}[x_k] + B_{du}[d_k])u(k) + B_d d(k) \quad (16)$$
$$B_{xu}[x_k] = [B_{xu,1}x(k), B_{xu,2}x(k), \ldots, B_{xu,4}x(k)] \quad (17)$$
$$B_{du}[d_k] = [B_{du,1}d(k), B_{du,2}d(k), \ldots, B_{du,4}d(k)], \quad (18)$$

Training. The output variable for training is the inside zone temperature, i.e. T_{in}. To train the trees $\mathcal{T}_{\iota,j}$ we consider weather disturbances, external disturbances due to occupancy and equipments, and autoregressive terms of the inside room temperature, i.e.

$$\mathcal{X} = \{\mathsf{T}_{\text{in}}(k), \ldots, \mathsf{T}_{\text{in}}(k-\delta_x), u(k), d(k+\bar{\delta}_d), \ldots, d(k-\underline{\delta}_d)\}, \quad (19)$$

where δ_x and δ_d represent the orders of the auto-regressive terms, and we have chosen $\delta_u = 0$. The training dataset was generated by simulating the bilinear model with rule-based strategies for 10 months in 2007, while the testing dataset was generated for 3 weeks of January. For the training we chose $\underline{\delta}_d = N - j + 1$, $\delta_x = 6$, where N is the predictive horizon, and either $\bar{\delta}_d = j - 1$ in the case with perfect forecast knowledge or $\bar{\delta}_d = 0$ in the case without forecast knowledge.

Validation. In Figure 2 we validate the prediction accuracy for horizon $N = 1, \ldots, 6$ (i.e. 6-hour ahead) for 3 weeks of January using SA w/ forecast (i.e. with perfect knowledge of future disturbance), SA w/o forecast (i.e. without any knowledge of future disturbance), and nMJS and oMJS (of course without any knowledge of future disturbance). We also compare the validation results with a baseline approach to switching regression, i.e. the k-LinReg algorithm. The Normalized Root Mean Squared

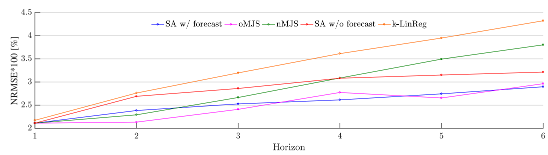

Fig. 2. Model validation on the *testing* dataset.

Error (NRMSE) behavior shows that, as expected, all RT-based methods provide exactly the same prediction quality at step 1, as the future disturbance has no effect. SA w/ forecast and oMJS always provides the best prediction, except for step 2 where the prediction of SA w/ forecast is slightly worse than nMJS due to model identification uncertainties. The oMJS and SA w/ forecast are instead comparable, since the optimized Markov chain can compensate the uncertainties induced by the disturbance on the model. nMJS works better than SA w/o forecast up to step 4: this shows that the stochastic information introduced by the Markov Chain identification process improves the quality of our prediction. However, after step 4, SA w/o forecast works better than nMJS: our interpretation is that, after 4 hours, the prediction error introduced by the Markov Chain model grows significantly making the overall accuracy worse than the case without any knowledge of future disturbance. This does not happen in the case of the optimized Markov chain. Also, the RT-based models (both deterministic and stochastic) outperform the predictive models obtained by means of the k-LinReg method.

Closed-loop simulations. Our objective is minimizing the energy usage, i.e. $c^\top u$, while maintaining a desired level of occupant comfort. The solution obtained from MPC with the bilinear model sets the optimal benchmark, since it uses the exact knowledge of the plant nonlinear dynamics and of the future disturbances. In what follows, we will call this solution the *oracle* where, at time step k, we solve the following continuously linearized MPC problem to determine the optimal sequence of inputs u^*:

Problem 6.

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \sum_{j=1}^{N-1} \Big(\left(x_{k+j} - x_{\mathsf{ref}}\right)^\top Q \left(x_{k+j} - x_{\mathsf{ref}}\right) \\ & \qquad + u_{k+j-1}^\top R u_{k+j-1} + c^\top u_{k+j-1} + \lambda \varepsilon_j \Big) \\ \text{subject to} \quad & x_{k+j} = A x_{k+j-1} + B u_{k+j-1} + B_d d_{k+j-1} \\ & B = B_u + B_{xu}[x_k] + B_{du}[d_{k+j-1}] \\ & x_{k+j} \in [x_{\mathsf{min}} - \varepsilon_j, x_{\mathsf{max}} + \varepsilon_j], \varepsilon_j \geq 0, \\ & u_{k+j-1} \in [u_{\mathsf{min}}, u_{\mathsf{max}}] \\ & x_k = x(k),\ j = 1, \ldots, N-1, \end{aligned}$$

where $Q \in \mathbb{R}^{12\times 12}$, $R \in \mathbb{R}^{4\times 4}$, $c^\top \in \mathbb{R}^{1\times 4}$ is proportional to the cost of using each actuator, and λ penalizes state bound violations ε_j. At each time step, only the first optimal input of the sequence is applied to the system. In

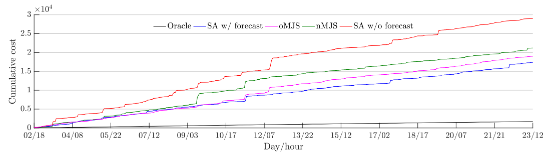

Fig. 3. Cumulative cost over 3 weeks of January.

Figures 1 and 3, we simulate the closed-loop system where the controller solves the MPC Problem 6 (considering the expected value in the stochastic cases) with prediction horizon $N = 6$ (i.e. 6 hours) for 5 different cases: SA w/ forecast, SA w/o forecast, nMJS, oMJS, and finally the *oracle*. The performance is compared for 3 weeks of January. The cooling usage factor C is constrained in $[0, 1]$, the heat input in $[0, 23]$ W/m^2, and the room temperature in $[21, 23]$ °C during the day. The optimization is solved in MATLAB using CPLEX. The reference temperature x_{ref} is chosen to be 22 °C. The cost function parameters are setup as $q_{11} = 10^2$, $R = diag(10^{-3})$, $\lambda = 10^3$, and $c^\top = [0,\ 3.32,\ 7.47,\ 1.107]$ as a constant cost of the electricity taken from (Gyalistras and Gwerder, 2010). The room temperature profile is shown in Figure 1. The plots show that oMJS and SA w/ forecast are the ones that provide less spikes, and are closer to the smooth temperature regulation of the *oracle*. We recall that the sampling time of our system is 1 hour, which explains the spiky behaviour of the temperature plot. The optimized cost function is shown in Figure 3. The plots show that, SA w/ forecast provides the best control performance, followed by oMJS that is quite close, nMJS, and finally SA w/o forecast. The *oracle* we compare to shows the best achievable control performance.

6. CONCLUSIONS AND FUTURE WORK

This paper provides a novel technique to identify a stochastic switching affine model from a dataset, combining RT with ARX system identification: this represents a further step towards bridging machine learning to control theory. Our novel modeling framework based on Markov Jump System models allows, when no disturbance forecast is available, to obtain prediction accuracies and control performance that are comparable to the case where perfect knowledge of the future disturbance is assumed. As an additional contribution, our framework allows to formally define and characterize important properties, e.g. such as stochastic stability and stabilizability. This will be investigated in future work leveraging preliminary results in De Iuliis et al. (2020). We also plan to validate our techniques to real experimental setups, and to validate our models on control systems where modeling a disturbance characterized by fast dynamics with a Markov Chain can be even more effective, e.g. when the disturbance is a communication channel in a networked control system.

7. ACKNOWLEDGEMENTS

This work was supported by the Italian Government under Cipe resolution n.135 (Dec. 21, 2012), project INCIPICT, and by the project VALU3S (call: H2020-ECSEL-2019-2-RIA). The authors would like to thank Xiaojing Zhang for providing the bilinear building model.

REFERENCES

Behl, M., Smarra, F., and Mangharam, R. (2016). Dr-advisor: A data-driven demand response recommender system. *Applied Energy*, 170, 30–46.

Bernardini, D. and Bemporad, A. (2012). Stabilizing model predictive control of stochastic constrained linear systems. *IEEE Transactions on Automatic Control*, 57(6), 1468–1480.

Breiman, L. (2017). *Classification and regression trees.* Routledge.

Costa, O.L.V., Fragoso, M.D., and Marques, R.P. (2006). *Discrete-time Markov jump linear systems.* Springer Science & Business Media.

De Iuliis, V., D'Innocenzo, A., Germani, A., and Manes, C. (2020). On the stability of discrete-time linear switched systems in block companion form. In *IFAC 2020 - 21st IFAC World Congress, Berlin.*

Di Girolamo, G., Smarra, F., Gattulli, V., Potenza, F., Graziosi, F., and D'Innocenzo, A. (2020). Data-driven optimal predictive control of seismic induced vibrations in frame structures. *Structural Control and Health Monitoring*, 27(4).

Gyalistras, D. and Gwerder, M. (2010). Use of weather and occupancy forecasts for optimal building climate control (opticontrol): Two years progress report main report. *Terrestrial Systems Ecology ETH Zurich R&D HVAC Products, Building Technologies Division, Siemens Switzerland Ltd, Zug, Switzerland*, 83.

Jain, A., Smarra, F., Behl, M., and Mangharam, R. (2018). Data-driven model predictive control with regression trees-An application to building energy management. *ACM Transactions on Cyber-Physical Systems*, 2(1), 4.

Jain, A., Smarra, F., and Mangharam, R. (2017). Data predictive control using regression trees and ensemble learning. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, 4446–4451. IEEE.

Lauer, F. (2013). Estimating the probability of success of a simple algorithm for switched linear regression. *Nonlinear Analysis: Hybrid Systems*, 8, 31–47.

Lun, Y.Z., D'Innocenzo, A., Smarra, F., Malavolta, I., and Di Benedetto, M.D. (2019). State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, 149, 174–216.

Merkblatt, S. (2006). 2024: Standard-nutzungsbedingungen für die energie-und gebäudetechnik. *Zürich: Swiss Society of Engineers and Architects.*

Oldewurtel, F. (2011). *Stochastic model predictive control for energy efficient building climate control.* Ph.D. thesis, ETH Zurich.

Smarra, F., Di Girolamo, G.D., De Iuliis, V., Jain, A., Mangharam, R., and D'Innocenzo, A. (2020). Data-driven switching modeling for mpc using regression trees and random forests. *Nonlinear Analysis: Hybrid Systems*, 36, 100882.

Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D'Innocenzo, A., and Mangharam, R. (2018a). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy.*

Smarra, F., Jain, A., Mangharam, R., and D'Innocenzo, A. (2018b). Data-driven switched affine modeling for model predictive control. In *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS'18)*, 199–204. IFAC.